

Opinnäytetyö (AMK)

Tietojenkäsittely

Yrityksen tietojärjestelmät

2015

Teemu Suoranta

SISÄLTÖRAKENTEIDEN SUUNNITTELU, TOTEUTUS JA HYÖDYNTÄMINEN WORDPRESS-YMPÄRISTÖSSÄ



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

Teemu Suoranta

SISÄLTÖRAKENTEIDEN SUUNNITTELU, TOTEUTUS JA HYÖDYNTÄMINEN WORDPRESS- YMPÄRISTÖSSÄ

Tämän opinnäytetyön tavoitteena on suunnitella, toteuttaa ja hyödyntää sisältörakenteita WordPress-sisällönhallintajärjestelmässä liiketoimintanäkökulmasta. Opinnäytetyön toimeksiantajana on Kalustetalo R-Top Oy. Opinnäytetyö on osa laajempaa projektia, johon liittyy sisältörakenteiden lisäksi graafinen suunnittelu ja käyttökoulutus. Projektin lopputuloksena syntyy verkkosivusto.

Työn teoriaosa käsittelee sisältörakenteiden suunnittelua ja WordPressin tietoarkkitehtuuria. Lähdemateriaalina on käytetty yleisesti sisältöarkkitehtuuria käsitteleviä teoksia, WordPressin dokumentaatiota sekä WordPress-kehittäjille suunnattuja teoksia. Teoriaosa pohjustaa sisältörakenteiden suunnitteluperiaatteet ja lähtökohdat WordPressin hyödyntämiselle sisällönhallintajärjestelmänä.

Empiirinen osa koostuu sisältörakenteiden määrittämisestä, toteutuksesta ja hyödyntämisestä. Työssä esitellään sisältölähtöinen prosessi verkkosivuston suunnitteluun ja toteutukseen. Esitettyä prosessia voidaan hyödyntää pohjana web-kehitysprojekteissa, jotka vaativat sisältörakenteiden suunnittelua.

ASIASANAT:

WordPress, sisältörakenteet, tietoarkkitehtuuri, sisältötyypit

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Business Information Technology | Business Information Systems

2015 | 38 pages

Tuomo Helo

Teemu Suoranta

DESIGNING, IMPLEMENTING AND UTILIZING CONTENT STRUCTURES IN WORDPRESS

The objective of this thesis was to design, implement and utilize content structures in WordPress for business needs. The client of the thesis was Kalustetalo R-Top Oy. The thesis was part of a broader project which includes graphical design and instructions. The final result of this project was a web site.

The theoretical part of the thesis consists of designing the content structures and description of the information architecture of WordPress. Source material includes general literature of content structure design, WordPress documentation and guides aimed for WordPress developers. The theoretical part underlays the design principles of content structures and the baseline for utilizing WordPress as a content management system.

The empirical part consists of defining the required content structures, implementing these structures and utilizing them. The thesis presents a content-oriented process for designing and building a website. The process can be utilized in projects that require designing of content structures.

KEYWORDS:

WordPress, content structures, information architecture, content types

SISÄLTÖ

1 JOHDANTO	6
2 SISÄLTÖRAKENTEIDEN SUUNNITTELU	8
2.1 Käyttäjäprofiilit	8
2.2 Sisällön luokittelu	10
2.3 Sivuston rakenne	11
3 WORDPRESSIN TIETOARKKITEHTUURI	15
3.1 Tietokanta-arkkitehtuuri	15
3.2 Hakemistorakenne	17
3.3 Toiminnallisuudet	19
3.4 Kritiikki	20
4 SISÄLTÖRAKENTEIDEN TOTEUTUS	22
4.1 Sisältötyypit	23
4.2 Kentät	23
4.3 Relaatiot	25
4.4 Hallintanäkymä	26
5 SISÄLTÖRAKENTEIDEN HYÖDYNTÄMINEN	28
5.1 Sivupohjat	28
5.2 Tuotesivu	29
5.3 Tuotelistaus	30
5.4 Valmistajasivu	32
5.5 Hakutoiminto	33
6 POHDINTA	35
LÄHTEET	37

LIITTEET

Liite 1. Tuotesivujen lisätieto-osion toteutus.

KUVAT

Kuva 1. WordPress-asennuksen juurihakemisto.	17
Kuva 2. Kontekstiriippuvaisen silmukan käyttö sivupohjassa.	19
Kuva 3. Sisältötyyppien välisen relaation määrittäminen.	25
Kuva 4. Sisältötyyppien näkyminen hallintäkäytössä.	26
Kuva 5. Hakutulosten esittäminen.	34

KUVIOT

Kuvio 1. Rautalankamalli tuotesivusta.	12
Kuvio 2. Rautalankamalli etusivusta.	13
Kuvio 3. Toteutettavan sivuston sisältötyyppien relaatiot.	22

TAULUKOT

Taulukko 1. R-Topin sisältötyypit.	10
Taulukko 2. Tärkeimmät tietokantataulut sisällönhallintaan WordPress-järjestelmässä (WordPress.org 2015a).	15
Taulukko 3. Ytimen ulkopuolisen wp-content-hakemiston rakenne.	18
Taulukko 4. Tuotteiden mukautetut kentät.	24

1 JOHDANTO

Tämän opinnäytetyön tavoitteena on suunnitella ja toteuttaa sisältörakenteet WordPress-sisällönhallintajärjestelmälle sekä hyödyntää niitä toimeksiantajan verkkosivutoteutuksessa. Opinnäytetyön toimeksiantajana on Kalustetalo R-Top Oy. Työssä tarkastellaan sisältörakenteiden suunnittelua teoreettisella tasolla ja WordPressin tietoarkkitehtuuria, luodaan sisältörakenteet ohjelmoimalla ja työkalujen avulla sekä hyödynnetään luotuja sisältörakenteita verkkosivustossa. Opinnäytetyön näkökulma on sisältörakenteiden suunnitteleminen liiketoiminnan ja markkinoinnin tarpeisiin.

R-Top on helsinkiläinen kalustealan yritys, joka myy pääosin kotimaisten valmistajien kalusteita. R-Top tarvitsi verkkosivut, joihin haluttiin esille kattava tuotevalikoima kategorioittain ja valmistajittain järjestettynä. Tuotteista haluttiin erottaa omaksi kokonaisuudekseen suoraan liikkeestä myytävät outlet-tuotteet sekä tilattavat tuotteet. Ostamisen helpottamiseksi yhteydenottolomake ja yhteystiedot haluttiin selkeästi jokaiselle sivulle sekä tuotteiden yhteyteen. Muita sivustolle tulevia sisältöjä olivat kuvastot, ajankohtaiset uutiset sekä yhteystietosivu. Sisällön lisääminen ja muokkaaminen haluttiin helppokäyttöiseksi, joten sivuston alustaksi valittiin WordPress-sisällönhallintajärjestelmä.

Projektiin kuului tarvittavien sisältörakenteiden ja ulkoasun suunnittelu, sivuston toteutus, sisällönsyötössä avustaminen, käyttökoulutus sekä julkaisu. Sivuston kehitys tapahtui paikallisessa kehitysympäristössä, josta sivusto siirrettiin julkaisussa tuotantopalvelimelle. Tässä opinnäytetyössä keskitytään projektin vaiheisiin, joissa käsitellään sisältörakenteita.

WordPress on opinnäytetyön kirjoitushetkellä ylivoimaisesti maailman käytetyin sisällönhallintajärjestelmä ja sen suosio on yhä kasvussa (BuiltWith 2015). WordPress oli alkuperäisessä muodossaan blogialusta, mutta siitä on kehittynyt joustava ja helposti laajennettava avoimen lähdekoodin sisällönhallintajärjestelmä, jolla on aktiivinen kehittäjäyhteisö (Hedengren 2014, 1-2). WordPress tar-

joaa web-kehittäjälle sisäänrakennettuina ominaisuuksina mahdollisuuden mukautettuihin sisältörakenteisiin sekä laajan lisäosakirjaston (McCollin 2012, 7-8). Oletuksena WordPress tarjoaa sisältötyypeinä sivut ja artikkelit. Sisältörakenteeltaan WordPressin oletuksesta poikkeavien sivustojen rakentaminen edellyttää uusien sisältörakenteiden suunnittelua ja toteutusta niin, että rakenteiden hyödyntäminen on tehokasta ja tarkoituksenmukaista.

Yleistasolla opinnäytetyön lähestymistapana on tapaustutkimus (case study), jossa vastausta tutkimuskysymykseen etsitään tutkimalla yksittäistä tapausta. Tapaustutkimuksessa tutkija on tiedostava osa tutkittavaa tapausta ja tutkimus ei ole luonteeltaan neutraalia. Tavoitteena ei ole luoda yleistettäviä teorioita vaan kuvata tutkimuskohde ja tehdyt havainnot. (Gillham 2010, 1-5.)

Tarkempi tutkimuksellinen lähestymistapa on tapaustutkimusten alainen konstruktivinen tutkimusote, joka selvittää tutkimuskysymyksiä ratkaisemalla reaali maailman ongelmia. Ratkaisuja ei löydetä vaan tutkimuksen tuloksena luodaan uutta. Tutkimus pohjautuu teoreettiseen aineistoon ja sen empiiristä lopputulosta verrataan teoriaan. (Lukka 2014.) Työn teoreettinen viitekehys käsittelee sisältörakenteiden suunnittelua sekä WordPressin tietoarkkitehtuuria.

2 SISÄLTÖRAKENTEIDEN SUUNNITTELU

Sisältörakenteiden suunnittelu pohjautuu kontekstin, sisällön ja käyttäjien analysointiin. Konteksti sisältää liiketoimintatavoitteet, käytettävän teknologian ja saatavilla olevat resurssit. Sisällöstä selvitetään sisältötyypit, sisällön määrä ja tavoitteet. Käyttäjien analysointi selvittää kohdeyleisön rakennetta, tarpeita ja tiedonhakutapoja. (Usability.gov 2015a.)

Sisältörakenteiden suunnittelu koostuu

- sisällön luokittelusta rakenteiksi sekä kaavakuviksi
- tiedon esittämisen suunnittelusta
- navigoinnin sekä tiedonhaun suunnittelusta (Crawford 2015a).

2.1 Käyttäjäprofiilit

Käyttötapausten tunnistaminen ja optimointi on osa sisältörakenteen suunnittelua. Käyttötapaukset luokitellaan erilaisten käyttäjäprofiilien avulla, jotka kuvaavat erityyppisiä kävijöitä. Käyttäjäprofiilit voidaan tunnistaa analysoimalla sivuston kävijöitä ja yrityksen asiakkaita tai määrittelemällä tavoiteasiakkaiden tarpeita ja piirteitä. Sisältörakenne voidaan rakentaa palvelemaan tasapuolisesti useita käyttäjäprofiileja tai painottaa suunnittelu jonkin käyttäjäprofiilin eduksi. (TecEd 2015.)

Käyttäjäprofiilien tavoitteita ja mahdollisia askeleita tavoitteiden saavuttamiseksi analysoidaan. Profiilien päämäärät ja tavat niiden saavuttamiseksi voivat vaihdella. Eri profiileja ei välttämättä kannata johdattaa samoihin toimintatapoihin tai samoille navigaatiopoluille. Erilaisten käyttäjien tarpeiden ja päämäärien määrittäminen auttaa sisällön rakentamista valittujen tavoitteiden täyttämiseksi. Profiilit ovat aina pelkistettyjä eivätkä todellisuudessa edusta ketään yksittäistä asiakasta. (TecEd 2015.)

Toteutettavan sivuston käyttäjäprofiileiksi määritettiin neljä erityyppistä asiakasta, jotka ovat

- yleisesti valikoimaa selaava asiakas
- tietyn valmistajan kalusteita etsivä asiakas
- tietyntyypistä kalustetta etsivä asiakas
- ensisijaisesti liikkeessä asioiva asiakas.

Yleisesti valikoimaa selaavalle pitää toteuttaa helposti selattavina listauksina koko tuotevalikoima kuvineen. Listauksia pitää olla useita ja järjestystä pitää voida muuttaa esimerkiksi hinnan tai aakkosjärjestyksen mukaan. Valituista käyttäjäprofiileista tämä on oletettavasti epätodennäköisin ostaja, sillä vierailulla ei ole selvää päämäärää. Tätä asiakastyyppeä pitää aktivoida ilmoittamalla ajankohtaisista tapahtumista, uusista kalusteista sekä kampanjoista etusivulla.

Tietystä tuotemerkestä kiinnostunut haluaa helposti nähdä kaikki jonkin valmistajan tuotteet. Jokaiselle valmistajalle on luotava oma sivu, joka kokoaa kaikki kyseisen valmistajan kalusteet ja kuvastot sekä tietoa valmistajasta. Jokaisen tuotteen yhteydessä pitää mainita valmistaja sekä esittää linkki valmistajakohtaiselle sivulle, sillä tälle asiakastyypille tuotemerkki tuo lisäarvoa.

Tietyntyyppeisiä kalusteita etsivä on kiinnostunut tietyn kriteerein rajatusta ryhmästä kalusteita. Kalustevalikoima voidaan ryhmitellä tuoteryhmien avulla, jotka ovat hierarkkisia. Esimerkiksi sohvat-tuoteryhmän alakategorioita ovat vuode- ja kulmasohvat. Navigoinnin avuksi tarvitaan myös hakutoiminto, jonka avulla voi löytää kalusteita, valmistajia ja kuvastoja.

Ensisijaisesti liikkeessä asioivalle asiakkaalle tärkeintä on aukioloaikojen ja osoitteen nopea löytyminen. Mobiililaitteilla aukioloajat tulee löytää helposti. Tämä asiakastyyppeä on kiinnostunut liikkeestä löytyvästä tuotevalikoimasta. Kalustetallolla on kahdentyyppisiä kalusteita, jotka ovat tilattavat (myöhemmin tuote) ja suoraan liikkeestä hankittavat (myöhemmin outlet-tuote) kalusteet. Outlet-tuotteet erotetaan omaksi kokonaisuudekseen, jotta tämä asiakasprofiili voi katsoa helposti liikkeessä olevia tuotteita verkkosivuilta ja tulla liikkeeseen kokeilemaan niitä.

Sivuston liiketoimintakontekstissa jokaisen käyttäjän toivottu toiminta on yhteydenotto tai käynti liikkeessä. Lopullinen tavoite on kaupan syntyminen, mutta rakennettavan järjestelmän kannalta lopullinen tavoite on yhteydenotto. Yhteystiedot ja yhteydenottolomakkeet tulee tehdä helposti jokaisen käyttäjäprofiiliin saataville.

2.2 Sisällön luokittelu

Sisältö luokitellaan sisällönhallintajärjestelmää varten sisältötyypeiksi, joiden alaiset yksittäiset edustajat (myöhemmin instanssi) sisältävät samantyyppistä tietoa. Sisältötyypit luodaan aiemmin määritettyjen käyttötapausten ja käyttäjäprofiilien tarpeisiin. Luokittelussa on otettava huomioon sisältötyyppien rooli ja esittäminen sivustolla. (Rueping 2010, 33-36.)

Sisältötyyppien sisältämä tieto jaetaan pieniin osiin. Nämä osat määritetään tiedonsyöttökentiksi, joiden avulla sisältötyyppien instanssien tieto tallennetaan. Osa sisältötyyppiin liittyvästä tiedosta voidaan esittää relaatioviittauksilla toisiin sisältötyyppeihin. Tarvittavien kenttien löytäminen pohjautuu jälleen käyttötapausten analysointiin. Kartoitetaan mitä tietoa kävijä tarvitsee toimintansa tueksi tai mikä tieto auttaa käyttäjää löytämään etsimänsä. (Laja 2013.)

Taulukko 1. R-Topin sisältötyypit.

Sisältötyyppi	Kuvaus
Valmistaja	Tuotemerkki, joka sisältää yrityksen nimen, logon ja lyhyen kuvauksen.
Tuote	Tilattava kaluste, jolla on nimi, kuva ja kuvaus. Relatio valmistajaan.
Outlet-tuote	Liikkeessä myytävä kaluste, jolla on nimi, kuva, kuvaus ja hinta. Relatio valmistajaan.
Tuoteryhmä	Ryhmä tuotteita tai outlet-tuotteita, jotka luokitellaan otsikolla.
Kuvasto	Koostuu nimestä, kansikuvasta ja PDF-muotoisesta tiedostosta. Relatio valmistajaan.
Uutinen	Yksinkertainen artikkeli, joka sisältää otsikon ja tekstialueen.

R-Topin sivustolle määritettiin taulukon 1 mukaiset sisältötyypit, joiden avulla useat sivuston sisäiset linkitykset ovat mahdollisia. Luokittelu mahdollistaa tuotteiden listaamisen valmistajittain, tuoteryhmittäin ja tuotetyypeittäin. Kaikki tieto valmistajista säilytetään kootusti omassa sisältötyypissään, jolloin esimerkiksi yksittäisen tuotteen valmistajan nimi voidaan noutaa valmistaja-sisältötyypistä eikä tietoa toisteta järjestelmässä. Taulukossa listattujen sisältötyyppien lisäksi tarvitaan sisältöä, joka ei perustu monistettaviin sisältötyyppeihin. Tällaista tietoa ovat aukioloajat, yhteystiedot ja etusivun nostot.

2.3 Sivuston rakenne

Sivuston organisaatorakennetta voi mallintaa hiearkkisella kaaviolla, sekvenssi-kaaviolla ja matriisilla. Hierarkiamallissa käyttäjä aloittaa navigoinnin yleisen tason sisällöstä ja liikkuu alasivuilla aina spesifimpään tietoon. Sekvenssikaavio antaa suoran toimintarakenteen, jossa mahdollisia reittejä loppupisteeseen on vain yksi. Matriisissa sisältö on useilla tavoilla linkittynyt sisäisesti ja käyttäjä voi saavuttaa tietyn sisällön useilla tavoilla. Organisaatorakenteita voidaan käyttää kuvaamaan koko sivuston rakenne, tietty toimintapolku tai osa sivustosta. (Usability.gov 2015b.)

Kestävä rakenteen suunnittelu huomioi sisällön määrän sekä skaalautuvuuden. Tehdyillä rakennevalinnoilla on pitkäaikaiset seuraukset sivuston toimivuudelle, helppokäyttöisyydelle sekä suorituskyvylle. Hyvä sivuston rakenne mahdollistaa sisältötasojen lisääminen sekä päätasoille että alitasoiksi. Hyvä organisaatorakenne ei ole liian syvä eikä liian matala. Liian syvä tasorakenne vie sisällön monen tason taakse ja tekee navigoinnista hankalaa. Liian matala tasorakenne vaatii valtavia valikkorakenteita ja tekee navigoinnista jälleen hankalaa. Sopiva tasapaino organisaatorakenteen syvyydelle tulee löytää suunnitteluvaiheessa sekä rakenteen helppokäyttöisyyttä tulee tarkkailla järjestelmän ylläpitovaiheessa. (Usability.gov 2015b.)

Rautalankamallinnus (wireframing) kuvaa sisällön sijoittamista sivustolla. Malli esittää pohjapiirroksen sisältörakenteen toteuttamiseen ilman visuaalista tyyliä.

Mallin avulla sivuston helppokäyttöisyyttä ja navigoinnin helppoutta voidaan arvioida ennen graafisen ilmeen suunnittelua. Rautalankamalli voi olla suunnittelu-prosessissa ennen graafista suunnittelua, prototyyppiä tai toteutusta. Rautalankamalli voidaan toteuttaa kuvana tai HTML:llä, jolloin mallissa voidaan esittää interaktiivisuutta ja responsiivisuus. (Lim 2012.)

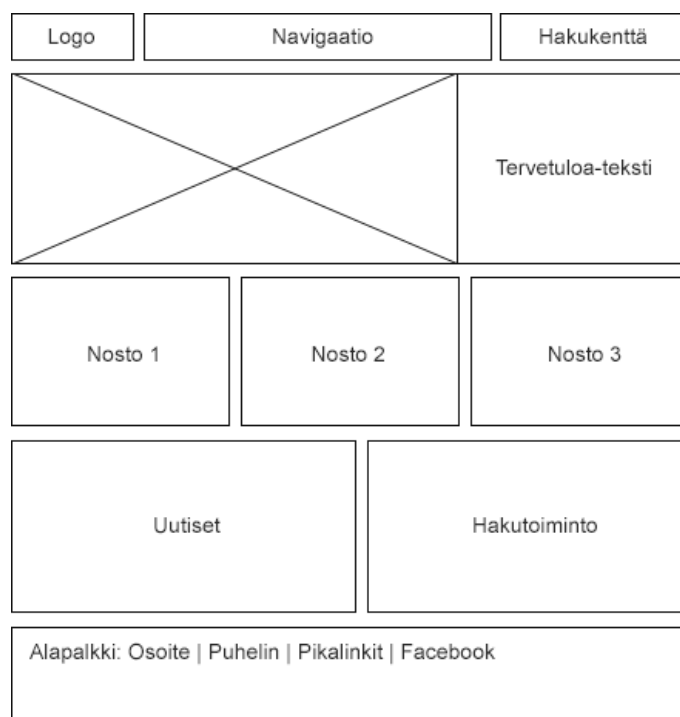
Logo	Navigaatio		Hakukenttä
Tuote-ryhmät	Tuotteen nimi		Tuoteryhmä
	Tuotekuva	Tuotekuvaus	
	Mitat Valmistaja Kysy tuotteesta		
Aukiolo-ajat	Katso myös (3kpl tuotteita)		
Alapalkki: Osoite Puhelin Pikalinkit Facebook			

Kuvio 1. Rautalankamalli tuotesivusta.

Toimeksiannon sivuston tuote-sivusta toteutettiin kuviossa 1 näkyvä rautalankamalli. Sivun rakenteessa navigaatioelementit löytyvät vasemmasta laidasta sekä yläpalkista. Yläpalkin navigaatio-alue sisältää logon, jota klikkaamalla pääsee etusivulle, ylätasen linkit tuotteet-, outlet-, valmistajat-, kuvastot- ja yhteystiedot-sivuille sekä hakutoiminnon. Yläpalkki toistuu kaikilla sivuilla. Vasemmassa laidassa on tuoteryhmät sekä aukioloajat, jotka ovat esillä kaikissa tuote- valmistaja- ja kuvastosivuilla. Kaikkien sivujen alalaidassa oleva alapalkki sisältää liikkeen osoitteen, puhelinnumeron ja sähköpostiosoitteen sekä pikalinkit tuoteryhmiin ja valmistajiin sekä Facebook-sivun linkin.

Tuotesivun sisältöosa koostuu tuotteesta, lisätiedoista ja kolmesta ehdotetusta lisätuotteesta. Tuotteen yläosa koostuu tuotteen nimestä sekä tuotteeseen liitetystä tuoteryhmistä. Tuotteella on yksi tai useita kuvia sekä tuotekuvaus. Tuotekuvausosio sisältää vapaata tekstiä sekä hinnan ja mahdollisen alennuksen. Lisätuotteet näytetään satunnaistetusti ja niiden valinta perustuu tarkasteltavan tuotteen tuoteryhmiin.

Tuotteen lisätiedot sisältävät kolme osiota, jotka aukeavat klikkaamalla. Oletuksena auki on mitat-osio, joka näytetään vain jos tuotteeseen on liitetty mittoja. Toinen osio on valmistaja, johon haetaan relaation avulla valmistajan nimi, logo ja kuvaus sekä linkki valmistajakohtaiselle sivulle. Viimeinen osio on kysy tuotteesta, joka sisältää yhteydenottolomakkeen sekä yhteystiedot.



Kuvio 2. Rautalankamalli etusivusta.

Etusivun rautalankamalli on kuvattu kuviossa 2. Ylä- ja alapalkin lisäksi etusivulla on tervetuloa-, nosto-, uutiset- ja hakutoimintoalueet. Tervetuloa-alue sisältää suuren kuvan sekä lyhyen esittelytekstin. Nostot ovat vapaavalintaisia poimintoja

sivustolta, jotka koostuvat kuvasta, otsikosta, tekstistä ja linkistä. Poimintojen avulla voidaan esitellä esimerkiksi tiettyä valmistajaa, tuoteryhmää tai kalustetta.

Uutiset-alue koostuu uutiset-sisältötyypin instansseista, joista näytetään viisi uutista ja linkki koko uutiseen. Uutiset-osiossa on myös linkki uutisarkistoon. Muut sivuston näkymät mukailevat näitä kahta rautalankamallia.

3 WORDPRESSIN TIETOARKKITEHTUURI

3.1 Tietokanta-arkkitehtuuri

WordPressin asennus luo MySQL-tietokantaan 11 tietokantataulua. Tauluihin syötetään asennuksen yhteydessä järjestelmän oletusarvot, asennusprosessissa syötetyt tiedot sekä testidataa. Asennuksessa luodaan käyttäjä, jolla on pääkäyttäjän oikeudet. Taulukossa 2 on esitetty sisällönhallinnan kannalta olennaisimmat taulut sekä taulujen tietosisältö. Tietokantataulujen etuliite on oletuksena "wp_", mutta sitä on mahdollista muuttaa asennusvaiheessa. (McCollin 2012, 56-59.)

Taulukko 2. Tärkeimmät tietokantataulut sisällönhallintaan WordPress-järjestelmässä (WordPress.org 2015a).

Tietokantataulu	Tietosisältö
wp_posts	Sisältötyyppien instanssit: artikkelit (post), sivut (page), tiedostoviitaukset (attachment), versiot (revision), navigaatioiden jäsenet (nav_menu_item) ja mukautetut sisältötyypit (custom post type)
wp_postmeta	Sisältötyyppien instanssien metadata
wp_term_relationships	Taksonomioiden ja sisältötyyppien instanssien relaatiot
wp_term_taxonomy	Taksonomiatyypit (kategoria, avainsana tai linkki)
wp_terms	Taksonomiatyyppien instanssit (termit)

Kaikilla sisältötyypeillä on wp_posts-taulussa samat sarakkeet, jotka määrittävät kaikkia sisältötyyppejä koskevat kentät. Mukautetut kentät (custom fields) tallennetaan wp_postmeta-tauluun. Mukautettuja kenttiä ei liitetä kiinteästi sisältötyyppiin vaan kenttiä on mahdollista ottaa käyttöön instanssikohtaisesti. Lisäosan avulla sisältötyyppien editoriin on mahdollista rajata tietyt mukautetut kentät, mutta tietokanta-arkkitehtuuri ei pakota samoja kenttiä kaikille sisältötyypin instansseille. (Ratnayake 2013, 78-80.)

Taksonomiat ovat sisältörakenne, joiden tarkoitus on ryhmitellä sisältötyyppien instansseja. Asennuksessa luotavat oletustaksonomiat ovat kategoriat (category), avainsanat (post_tag) ja linkkien kategoriat (link_category), joka on poistunut käytöstä WordPressin versiossa 3.5. Mukautettuja taksonomioita (custom taxonomy) on mahdollista luoda ja liittää useaan sisältötyyppiin. Taksonomiatermien ja sisältötyyppien instanssien väliset yhteydet tallennetaan wp_term_relationships-tauluun. (WordPress.org 2015b.)

Sisältötyypit ja taksonomiat voivat olla hierarkkisia. Oletuksena hierarkiaa tukevat sisältötyypeistä sivut, versiot, navigaation jäsenet, liitetiedostot sekä taksonomioista kategoriat ja linkkien kategoriat. Teemat ja lisäosat kykenevät muuttamaan oletussisältörakenteiden hierarkiatukea. Hierarkian avulla saman sisältötyypin instansseja voi linkittää toisiinsa sekä näyttää oletuskenttiä ja tehdä kyselyitä perustuen instanssin hierarkia-asemaan. (WordPress.org 2015c.)

Käyttäjien oikeuksien hallinta tapahtuu WordPressissä roolien avulla. Käyttäjät tallennetaan wp_users-tauluun, joista on viittaus wp_usermeta-taulun rooliin. Roolit on kuvattu wp_options-taulussa. Järjestelmän asennuksessa luotu käyttäjä saa pääkäyttäjän roolin. Roolien avulla käyttäjiltä voidaan rajata oikeuksia julkaista, muokata tai poistaa sisältöä sekä muuttaa järjestelmän asetuksia. (WordPress.org 2015d.)

Sisältötyypit saavat automaattisesti rekisteröinnissään WordPressin ytimen kyvyn suorittaa create-, read-, update- ja delete-operaatioita tietokantaan (Ratnayake 2013, 94). Käyttäjä- ja käyttäjäroolitasoilla vastaavia oikeuksia voidaan poistaa (Ratnayake 2013, 39).

Tietokantaan tallennetaan viittaukset kuviin sekä muihin järjestelmiin ladattuihin tiedostoihin. Tiedostoviittaukset ovat oma sisältötyyppinsä (attachments) ja niihin voidaan kohdistaa kyselyitä samaan tapaan kuin muihin sisältötyyppeihin. Jos tiedosto ladataan toisen sisältötyypin yhteydessä kuten artikkelin instanssin editorissa, lisätään tiedostoviittauksen yläsivuksi (parent) muokattavan instanssin ID. Tiedostoviittauksista tallennetaan metadatanä esimerkiksi kuvista mittasuhteet wp_postmeta-tauluun. (WordPress.org 2015e.)

3.2 Hakemistorakenne

WordPressin asennuspaketista purettava ydin muodostuu pääasiassa PHP-tiedostoista. Kuvassa 1 esitetään hakemistorakenne WordPressin juurihakemistossa. Asennuspaketti ei sisällä erikseen SQL-muotoista tietokantapakettia vaan tarvittavat tietokantataulut luodaan PHP:n avulla asennuksessa. (Hedengren 2014, 27.)

wp-admin	Tiedostokansio	
wp-content	Tiedostokansio	
wp-includes	Tiedostokansio	
index.php	PHP-tiedosto	1 kt
license.txt	Tekstitiedosto	20 kt
readme.html	Chrome HTML ...	8 kt
wp-activate.php	PHP-tiedosto	5 kt
wp-blog-header.php	PHP-tiedosto	1 kt
wp-comments-post.php	PHP-tiedosto	5 kt
wp-config.php	PHP-tiedosto	3 kt
wp-cron.php	PHP-tiedosto	3 kt
wp-links-opml.php	PHP-tiedosto	3 kt
wp-load.php	PHP-tiedosto	3 kt
wp-login.php	PHP-tiedosto	33 kt
wp-mail.php	PHP-tiedosto	9 kt
wp-settings.php	PHP-tiedosto	11 kt
wp-signup.php	PHP-tiedosto	25 kt
wp-trackback.php	PHP-tiedosto	4 kt
xmlrpc.php	PHP-tiedosto	3 kt

Kuva 1. WordPress-asennuksen juurihakemisto.

Hakemistorakenteessa erotetaan toisistaan ydin (core) ja sivustokohtaiset tiedostot. Kaikki wp-content-kansion ulkopuolella olevat tiedostot ja hakemistot rajataan kuuluvaksi ytimeen. Ytimen ulkopuolisen kansion hakemiston rakenne on kuvattu taulukossa 3. Ytimeen kuuluvia tiedostoja ei suositella muokkaamaan, pois lukien .htaccess ja wp-config.php, sillä WordPressin päivitykset voivat yliajaa muutokset. Tiedosto wp-config.php sisältää järjestelmän konfiguraatioasetukset kuten tietokantatunnukset. Ytimen toimintaan on mahdollista vaikuttaa lisäosilla ja teemalla, jolloin WordPressin päivitykset eivät yliaja tehtyjä muutoksia. (Hedengren 2014, 27-29.)

Taulukko 3. Ytimen ulkopuolisen wp-content-hakemiston rakenne.

Hakemisto	Sisältö
languages	WordPressin ja sen lisäosien kieliversiot
plugins	Lisäosat
themes	Asennetut teemat
upgrade	Päivitysten väliaikainen tallennuspaikka
uploads	Kuvat sekä muut ladatut tiedostot

Teemat sisältävät sivupohjat, sivuston tyylitiedostot, toiminnallisuuksia sekä muita ulkoasussa tarvittavia tiedostoja. Teemojen sivupohjien määrä ja hakemistorakenteet eroavat toisistaan. Ytimen vaatimat tiedostot teemalta ovat index.php ja style.css. Järjestelmän themes-kansiossa voi olla useita teemoja asennettuna, mutta vain yksi teema voi olla kerrallaan aktiivisena. WordPressin asennuksen mukana tulee aina oletusteema. (Coyier & Starr 2014, 51-53.)

Lisäosat ovat teemasta ja ytimestä irrallisia moduuleita, jotka lisäävät järjestelmään toimintoja tai muokkaavat ytimen toimintaa. Lisäosien asennus tapahtuu joko manuaalisesti purkamalla lisäosan tiedostot plugins-kansioon tai lataamalla lisäosa WordPressin sisäänrakennetulla latausominaisuudella. Lisäosa voi olla aktiivisessa tai passiivisessa tilassa. Passiivisessa tilassa oleva lisäosa ei ole käytössä, mutta sen tiedostoja ei ole poistettu. WordPress ylläpitää tietoa lisäosien aktiivisuudesta tietokannassa wp_options-tilussa. (Coyier & Starr 2014, 131-134.)

Kuvat ja muut Media Managerilla ladatut tiedostot tallennetaan uploads-kansioon. Riippuen WordPressin asetuksista tiedostot järjestetään joko hakemiston juureen tai eroteltuna kuukausittaisiin alihakemistoihin. Kuvista skaalataan ja rajataan erikokoisia kuvia perustuen teeman ja lisäosien määrittämiin kuvakokoihin. Oletuksena WordPressissä on neljä kuvakokoa, jotka ovat thumbnail (150x150 rajauksella), medium (300x300 ilman rajausta), large (1024x1024 ilman rajausta) sekä alkuperäinen kuva. Jos alkuperäinen kuva on kohdekoko pienempi, uutta versiota ei luoda. (WordPress.org 2015f.)

3.3 Toiminnallisuudet

WordPressin toiminnallisuudet muodostuvat ytimestä, lisäosista ja teemasta. Ydin sisältää perustoiminnallisuudet tiedon tallentamiseen ja noutamiseen tietokannasta, hallintanäkymät ja käyttäjienhallinnan. Lisäosat laajentavat ja muuttavat ytimen toiminnallisuuksia. Teema määrittelee sivupohjat ja hyödyntää ytimen ja lisäosien funktioita sekä kykenee itse laajentamaan ja muokkaamaan ytimen toimintaa. (Król & Silver 2013.)

Teemojen käsittely ei noudata MVC-arkkitehtuuria (model-view-controller) eli logiikkaa ei eroteta näkymistä. Teemojen sivupohjat ovat sekoitus HTML-koodia ja PHP-logiikkaa. Tarvittaessa logiikan määrää voi rajoittaa teemassa siirtämällä näkymien loogisen käsittelyn lisäosille. (Ratnayake 2013, 12.)

Tietokantaan tallennettujen sisältötyyppien instanssien noutaminen sivupohjiin tapahtuu kontekstista riippuvalla silmukalla (The Loop), mukautetuilla kyselyillä (WP_Query) tai suorilla SQL-kyselyillä. Kontekstiriippuvaisen silmukan käyttö on esitetty kuvassa 2. Kyseinen silmukka tulostaa kyselyn kohteena olevan sisältötyypin instanssin otsikon ja sisällön, joka on wp_posts-tietokantataulun post_content-sarakkeen sisältö. (Hedengren 2014, 41-42.)

```
1  <?php get_header(); ?>
2
3  <?php if (have_posts()) : ?>
4      <?php while (have_posts()) : the_post(); ?>
5
6          <h1><?php the_title(); ?></h1>
7          <?php the_content(); ?>
8
9      <?php endwhile; ?>
10 <?php endif; ?>
11
12 <?php get_footer(); ?>
```

Kuva 2. Kontekstiriippuvaisen silmukan käyttö sivupohjassa.

WP_Query on ytimen määrittämä luokka, joka huolehtii tietojen noutamisesta tietokannasta. Kontekstiriippuvainen The Loop on WP_Queryn erikoistapaus.

WP_Query:n avulla voidaan luoda sisältökyselyjä, joissa tulosjoukkoa rajataan esimerkiksi sisältötyypin, liitetyn kategoriatermin ja kirjoittajan perusteella. WP_Query-luokalta on mahdollista palauttaa tietoa käsiteltävästä kyselystä. (WordPress.org 2015g.)

WordPressin oletusosoiterakenne viittaa URL-osoitteessa suoraan sisältötyypin instanssin indeksiin wp_posts-taulussa ja on muotoa <http://osoite.fi/?p=1>. Oletusosoitteesta voidaan ytimessä olevan toiminallisuuden avulla muuttaa selkokieliseksi versioksi kuten muodoksi <http://osoite.fi/sivun-nimi>. Osoitteen uudelleenkirjoitus hyödyntää palvelimen mod_rewrite- tai lighttpd-moduulia. (WordPress.org 2015h.)

3.4 Kritiikki

WordPress oli alkuperäisessä muodossaan blogialusta, josta kehittyi sisällönhallintajärjestelmä (McCollin 2012, 7). Tämä historia näkyy vahvasti tietorakenteiden nimeämiskäytännöissä, jotka eivät ole tämän hetkisessä versiossa enää kuvaavia. WordPress on aluksi toteutettu proseduurillisesti, minkä jälkeen kehitys on suuntautunut oliopohjaiseen toteutukseen. Tämän hetkisessä muodossaan järjestelmä on sekoitus vanhaa proseduurillista rakennetta ja uutta oliopohjaista toteutusta. (Ratnayake 2013, 22.)

Suuret sivustokokonaisuudet saattavat olla luontevampia ja suorituskykyisempiä toteuttaa modulaarisesti toteutetulla sisällönhallintajärjestelmällä kuten Drupalilla. Tämän hetkisellä WordPress-järjestelmällä useat suuremmatkin sivustokokonaisuudet ovat mahdollisia toteuttaa, mutta modulaaristen ominaisuuksien lisääminen edellyttää edistynyttä osaamista teeman ja lisäosien kehityksestä. (Hedengren 2014, 224.)

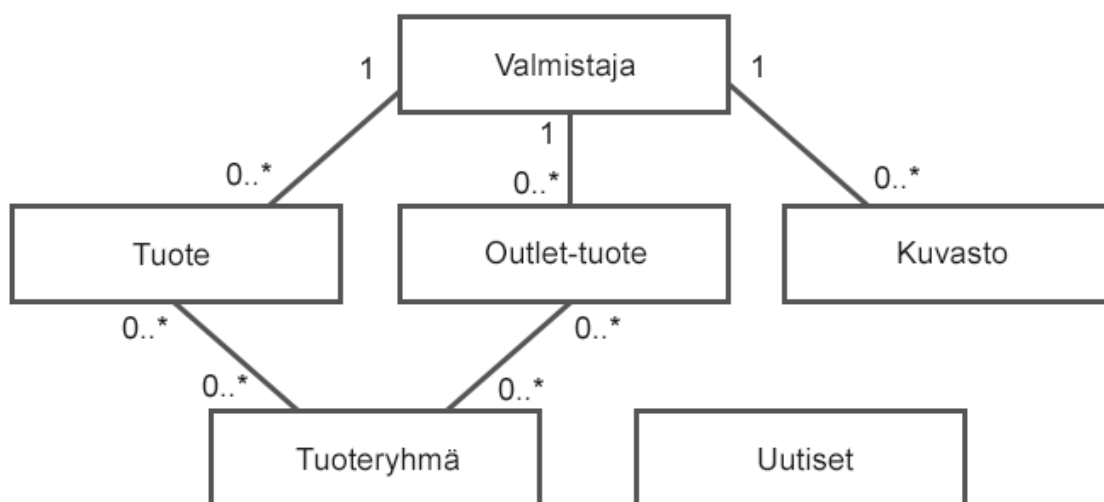
WordPressin tietoarkkitehtuuri on ensisijaisesti suunniteltu sisällönhallintajärjestelmän tarpeisiin. WP:n käyttö verkkosovelluksen pohjana saattaa tuoda esiin tietokantarakenteen rajoituksia skaalautuvuudelle. Eräs rajoite on WordPressin metataulujen runsas käyttö, mikä luo haasteita suorituskyvylle monimutkaisissa ky-

selyissä ja hakutoiminnallisuuksissa. Ongelman voi ratkaista käyttämällä mukautettuja tietokantatauluja. Oletuksena käytössä olevat versionhallinta ja automaattinen tallennus saattavat myös hidastaa verkkosovelluksen toimintaa täyttämällä tietokantaa. (Ratnayake 2013, 90-92.)

Verrattuna kaupallisiin sisällönhallintajärjestelmiin WordPressiltä puuttuu avoimen lähdekoodin järjestelmänä virallinen käyttötuki. Täten WordPress ei myöskään anna takuuta järjestelmän toiminnasta. WordPressin käyttötuki koostuu sisällönhallintajärjestelmän ulkopuolisista konsulteista ja web-kehittäjistä. WordPress.org-sivustolta löytyy tukifoorumi, jossa vapaaehtoiset vastaavat kysymyksiin ja avustavat ongelmissa. (Hedengren 2014, 225.)

4 SISÄLTÖRAKENTEIDEN TOTEUTUS

Tässä projektin vaiheessa sisältö rakenne on suunniteltu ja WordPress asennetaan paikalliseen kehitysympäristöön. Ympäristö on toteutettu WAMP:in avulla, joka on kokoelma ohjelmia, jotka muodostavat Apache-palvelimen, MySQL-tietokannan ja PHP-tulkkauksen. WAMP mahdollistaa sivuston rakentamisen tuotantopalvelinta mukailevissa olosuhteissa.



Kuvio 3. Toteutettavan sivuston sisältötyyppien relaatiot.

Toimeksiannon verkkosivuilta pystyttiin sisältö rakenteen suunnitteluvaiheessa rajaamaan erilaisia sisältötyyppejä, jotka on relaatioineen kuvattu kuviossa 3. Valmistaja-sisältötyyppiin liitetään tuotteita, outlet-tuotteita sekä kuvastoja. Kullakin valmistajalla voi olla useita tuotteita, outlet-tuotteita tai kuvastoja, mutta näillä voi olla vain yksi valmistaja. Tuotteiden ja outlet-tuotteiden ryhmittelyyn käytetään tuoteryhmiä. Tuoteryhmät ovat hierarkkisia ja tuote tai outlet-tuote voi kuulua yhtä aikaa moneen tuoteryhmään. Uutiset-tietotyyppi ei linkity relaatioilla muihin tietotyyppihin vaikka se voi sisältötasolla sisältää linkkejä muihin tietotyyppihin. Kuviossa 3 esitettyjen sisältötyyppien lisäksi tarvitaan WordPressin oletussisältötyypeistä sivuja etusivuun, tuotesivuun, uutisarkistoon ja yhteystietoihin.

4.1 Sisältötyypit

Sisältötyypit on jaettu WordPressissä julkaisutyyppeihin (post type) ja taksonomioihin (taxonomy). Julkaisutyypit ovat julkaistavaa sisältöä kuten artikkelit, tuotteet ja valmistajat. Taksonomiat ovat julkaisutyyppeiden luokitteluun tarkoitettu sisältörakenne, joita voivat olla esimerkiksi tuoteryhmä, avainsanat ja valmistusmaa. (Hedengren 2014, 232-233.)

Mukautettujen julkaisutyyppeiden rekisteröintiin voidaan käyttää ytimen funktiota `register_post_type` ja taksonomioiden rekisteröintiin funktiota `register_taxonomy`, jotka voidaan sijoittaa teeman `functions.php`-tiedostoon tai lisäosaan (Wordpress.org 2015c). Sisältötyyppien ja taksonomioiden rekisteröimiseen ja hallintaan käytetään tässä opinnäytetyössä lisäosaa Custom Post Type UI, joka tarjoaa WordPressin hallintapaneeliin graafisen käyttöliittymän.

Kuviossa 3 esitetyistä sisältötyypeistä kaikki toteutetaan mukautettuina julkaisutyypeinä lukuun ottamatta tuoteryhmiä, jotka toteutetaan taksonomiana. Kaikki julkaisutyypit määritettiin ei-hierarkkisina ja tuoteryhmä määritettiin hierarkkiseksi, koska tuoteryhmiä halutaan luoda useille tasoille. Tuoteryhmät liitettiin rekisteröinnin yhteydessä tuotteisiin ja outlet-tuotteisiin. Rekisteröinnin jälkeen julkaisutyypit näkyvät hallintanäkymän vasemmassa palkissa.

4.2 Kentät

Sisältötyyppikohtaiset kentät määritettiin Advanced Custom Fields -lisäosalla (myöhemmin ACF), joka hyödyntää WordPressin ytimen custom fields -rajapintaa. Mukautettujen kenttien data tallennetaan `wp_postmeta`-tietokantatauluun. ACF mahdollistaa useista kentistä koostuvien kenttäryhmien (field groups) luomisen. Kenttäryhmille voidaan määrittää säännöt, joiden perusteella kentät otetaan käyttöön sisältötyyppien instanssien muokkausnäkymissä. (Condon 2015.)

Kenttäryhmät tallennetaan oletuksena `wp_posts`-tietokantatauluun `acf-sisältötyyppinä`. Kenttien määrittely on tallennettu `wp_postmeta`-tauluun, johon viitataan

wp_posts-taulun pääavaimella. Kenttäryhmiä on mahdollista säilyttää myös teeman functions.php-tiedostossa tai lisäosassa, mikä mahdollistaa tuotantosivuston kenttien muokkauksen paikallisessa testiympäristössä ja muutosten hallitun käyttöönoton tuotantopalvelimella ilman tietokantamuutoksia.

Taulukossa 4 on esitetty tuotteille määritelty kenttäryhmä. Kyseinen kenttäryhmä liitettiin sekä tuotteisiin että outlet-tuotteisiin. Kenttien tunnukset ovat tuote-alkuliitteellä jotta kenttien tunnukset ovat varmasti yksilöllisiä. Kenttäryhmä kykenee piilottamaan sisältötyyppien instanssien muokkausnäkömäästä tarpeettomia ominaisuuksia. Tuotteilta piilotettiin muut oletusominaisuudet paitsi osoite ja kategoriat.

Taulukko 4. Tuotteiden mukautetut kentät.

Nimiö	Tunnus	Kenttätyyppi	Tarkoitus
Tuotteen kuvaus	tuote-txt	Wysiwyg Editor	Visuaalinen editori, jossa kuvataan yleisesti tuotetta.
Tuotteen mitat	tuote-mitat	Text Area	Tekstikenttä, jossa on tuotteen mitat.
Hinta	tuote-hinta	Number	Hinta ilman valuuttasymbolia.
Esittelykuva	tuote-kuva	Image	Kuva, joka näytetään tuotelistauksissa ja pääkuvana tuotesivulla.
Kuvagalleria	tuote-galleria	Repeater	Sisältää valinnaisen määrän kuvia.
Lisäasetukset	tuote-check	Checkbox	Näyttää tai piilottaa alennuksiin liittyvät kentät. Sisältää valinnan "Laita alennukseen".
Uusi hinta	tuote-ale	Number	Alennettu hinta. Oletuksena piilotettu.
Ale alkaa	tuote-alku	Date Picker	Alennusmyynnin alkupäivämäärä. Oletuksena piilotettu.
Ale loppuu	tuote-loppu	Date Picker	Alennusmyynnin loppumispäivä. Oletuksena piilotettu.

Muut luodut kenttäryhmät olivat valmistaja- ja kuvasto-sisältötyyppien kentät, etusivun ja yhteystietosivun kentät sekä yhteystiedot. Yhteystietokentät sidottiin ACF:n options-rajapintaan, jolloin kentät näkyvät hallintanäkymässä omana koh-

tanaan vasemmassa palkissa eivätkä ole liitettynä mihinkään sisältötyyppiin. Options-sivun kenttiä voidaan kutsua missä tahansa kontekstissa tai kyselyssä teeman sivupohjassa.

4.3 Relaatiot

Oletuksena WordPressin ydin tarjoaa kaksi tapaa yhdistää ja ryhmitellä sisältötyyppien instansseja. Taksonomiat ryhmittelevät instanssit ja hierarkia mahdollistaa instanssien määrittämisen eri tasoille. Suoraa tukea instanssien välisille yhteyksille sisältötyyppien sisällä ja välillä ei ole. Tämän ongelman ratkaisee lisäosa Posts 2 Posts (myöhemmin P2P). Lisäosa luo tietokantaan taulut wp_p2p ja wp_p2pmeta, joihin instanssien yhteyden tallennetaan. P2P mahdollistaa many-to-many- ja one-to-many-yhteyksien luomisen instanssien välille. One-to-one-rajoitus on mahdollista luoda käyttöliittymätasolla piilottamalla one-to-many-yhteyden toisen osapuolen yhteydenhallinta muokkaustilassa, jolloin käyttäjän on mahdollista tehdä vain one-to-one-tyyppisiä yhteyksiä. (Ratnayake 2013, 90-92.)

Edellä mainitut relaatiotavat ovat kaksisuuntaisia eli molemmista osapuolien kontekstista voidaan todeta relaation olemassaolo sekä selvittää toinen osapuoli. Yksisuuntaisessa relaatiossa vain toinen osapuoli voi saada tietoa relaation olemassaolosta. Yksisuuntaisen relaation tapoja ovat esimerkiksi sisäiset linkitykset tai ACF:n kenttä post object. Toteutettavassa sivustossa yksisuuntaisista relaatioista on käytössä vain sisäiset linkitykset, jotka lisätään manuaalisesti tekstieditorilla instanssien muokkaustilassa.

```

136
137 function my_connection_types() {
138     p2p_register_connection_type( array(
139         'name' => 'tuotemerkki_tuote',
140         'from' => 'tuotemerkki',
141         'to' => 'tuote',
142         'cardinality' => 'one-to-many',
143         'admin_dropdown' => 'any'
144     ) );
145 }
146 add_action( 'p2p_init', 'my_connection_types' );
147

```

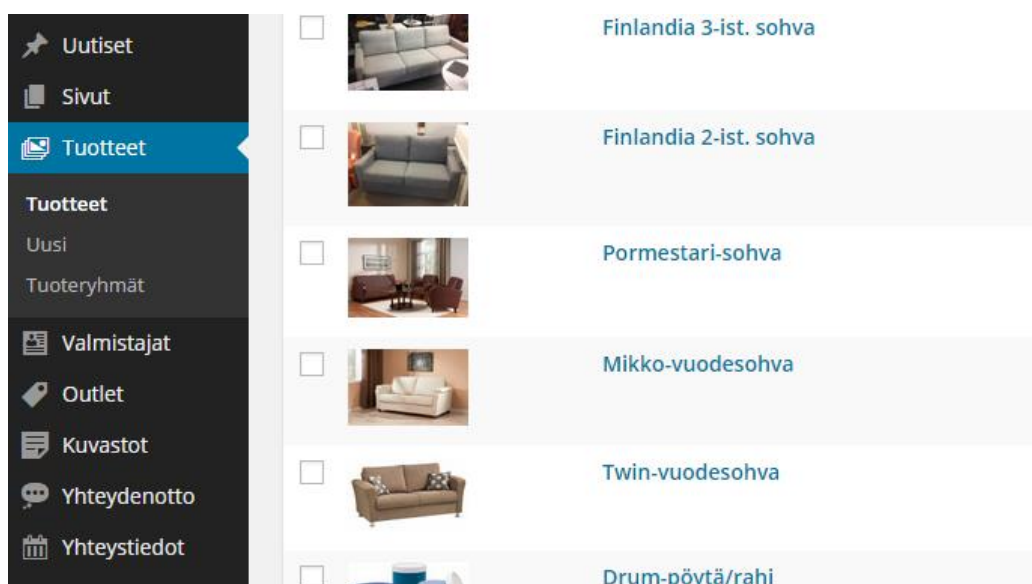
Kuva 3. Sisältötyyppien välisen relaation määrittäminen.

Tarvittavat yhteydet määritettiin toimeksiannon teeman functions.php-tiedostoon kuvan 3 mukaisesti. Yhteystyyppien rekisteröinnissä määritetään yhteyden nimi, osapuolien sisältötyyppien tunnukset, lukumääräsuhteet (cardinality) ja yhteyksien muodostaminen hallintanäkymässä (admin_dropdown).

Yhteyksiä luotiin kuvion 3 perusteella. Tuoteryhmien yhteys muodostettiin sisältötyyppien rekisteröinnissä kun taksonomia asetettiin voimaan tuotteiden ja outlet-tuotteiden sisältötyypeille. P2P:lle määritettiin kolme yhteystyyppiä, joiden kaikkien kohteena oli valmistaja-sisältötyyppi. Yhteydet määritettiin näkymään hallintanäkymissä molemmilla osapuolilla, jolloin yhteyden pystyi muodostamaan molempien osapuolien instanssien muokkausnäkymissä. Yhteydet olivat tyyppiä one-to-many.

4.4 Hallintanäkymä

Sisältötyyppien rekisteröinnin yhteydessä sisältötyyppien nimeämiskäytännöt sekä sijainnin hallintanäkymän vasemmassa sivupalkissa voi määrittää. Oletuksena kaikki julkaisutypit saavat saman ikonin kuin artikkelilla. Muuttamalla ikoneita ja järjestystä hallintakäyttökokemusta saadaan parannettua.



Kuva 4. Sisältötyyppien näkyminen hallintanäkymässä.

Hallintanäkymän navigaatorakennetta voi muokata Admin Menu Editor -lisäosan avulla. Luoduille julkaisutyypeille lisättiin kuvaavammat ikonit sekä luotu options-osio nimettiin Yhteystiedot-osioksi ja sen ikoni vaihdettiin. Oletusjulkaisutyyppi artikkelit piilotettiin hallintanäkymästä, sillä sivusto ei käytä artikkeleita. Kuvassa 4 esitetään hallintanäkymän vasemman palkin lopputulos.

Julkaisutyyppien instansseja voidaan selata hallintanäkymässä listana. Oletuksena tietosarakkeina ovat otsikko, kirjoittaja ja päivämäärä. Sarakkeita voi muuttaa ja niiden hyödyntäminen helpottaa instanssien selaamista ja oikean instanssin löytämistä.

Hallintanäkymän rivejä muutettiin Admin Columns -lisäosalla. Tuotteiden ja outlet-tuotteiden sarakkeiksi määritettiin kuva, otsikko, hinta, alennus ja tuoteryhmät. Kuvastojen sarakkeiksi määritettiin otsikko, alaotsikko ja kansikuva. Valmistajien sarakkeina näytetään otsikko ja logo.

5 SISÄLTÖRAKENTEIDEN HYÖDYNTÄMINEN

Tässä projektin vaiheessa tarvittavat sisällörakenteet on luotu ja hallintanäkymä on viimeistelty. Sisältötyyppeihin on syötetty oikeaa dataa. Seuraavaksi rakennetaan verkkosivuston kävijälle näkyvä osuus eli ulkoasu, sivupohjat ja tiedonhaku-logiikka. Tiedonsyöttö on mahdollista tehdä ennen tätä vaihetta. Tämän vaiheen jälkeen sivusto on valmis julkaistavaksi, mikäli sisältö arvioidaan riittäväksi. Tämä luku käsittelee vain olennaisimpia osia rakennettavasta sivustosta.

5.1 Sivupohjat

Sivupohjat määritetään WordPressissä teemaan. Projektille luodaan uusi tyhjä teema. Yksittäinen sivupohja muodostuu hyvien käytäntöjen mukaan aina useammasta osasta, jotka on tallennettu omiin tiedostoihinsa. Ainakin sivun otsaketiedot (`header.php`) ja alapalkki (`footer.php`) tulee olla erotettuna, jotta vältetään koodin toistuminen. WordPressissä on sisäänrakennetut funktiot `get_header` ja `get_footer` edellä mainittujen osien noutamiseen. Muita sivupohjan osia voidaan kutsua `get_template_part`-funktiolla. (Hedengren 2014, 67-70.)

WordPress valitsee kullekin sivulle näytettävän sivupohjan sivun URL-osoitteessa olevien parametrien perusteella. WordPressin dokumentaatiossa kuvataan sivupohjan valinta perustuen näytettävään sisältöön. Sivupohjan valinta lähtee aina spesifimmästä sivupohjasta kuten `front-page.php` ja jos spesifejä sivupohjia ei löydy päädytään lopulta pakolliseen `index.php`-sivupohjaan. Sivupohjan saa lisättyä järjestelmän sivupohjahierarkiaan nimeämällä sen WordPressin nimeämiskäytäntöjen mukaan. Sisältötyypille on mahdollista määrittää useita sivupohjia, joita voi valita instanssin muokkaustilassa. (WordPress.org 2015i.)

Teeman rakennuksessa hyödynnettiin Bootstrap-tyylikehystä, joka sisältää valmiiksi määriteltäviä CSS-luokkia ja JS-funktioita sekä ikonifontin. Tuotteiden esityslistausnäkymissä ja tuotteiden järjestely toteutettiin jQuery-kirjastoon perustuvalla Isotope-skriptillä. Nämä työkalut vaikuttavat sivupohjien lähdekoodin rakenteeseen ja nimeämiskäytäntöihin.

5.2 Tuotesivu

Tuotesivu toteutetaan sivupohjaan `single-tuote.php`, jolloin sitä käytetään näytettäessä tuote-sisältötyypin instanssi. Tuotesivun rakenne määritettiin suunnitteluvaiheessa rautalankamallilla kuviossa 1. Tässä alaluvussa keskitytään vain itse sisältöosan toteutukseen, joka sisältää tuotteen päätiedot, lisätiedot sekä lisätuotteet.

Sisältöosan tietoa noudetaan instanssista kontekstiriippuvalla silmukalla. Tuotteen nimi haetaan `the_title`-funktion avulla. Tuoteryhmät noudetaan `wp_get_post_terms`-funktiolla, joka palauttaa taulukon, joka sisältää instanssiin liitetyt taksonomiatermiä oliona. Taulukon sisältö käsitellään `foreach`-silmukalla ja termien nimet sekä tarvittava HTML-merkkaus tulostetaan.

Sisältöosion tuotokuva esitetään joko yksittäisenä kuvana tai kuvasarjana. Kuvaosion alussa tarkastetaan onko tuote-galleria-kenttään syötetty enemmän kuin yksi kuva. Jos kuvia on vain yksi, haetaan kuvan id, jonka perusteella muodostetaan URL-osoitteet sivulla näytettävälle kuvakoolle sekä suurelle kuvalle, joka näytetään kun kuvaa klikataan. Täysikokoisen kuvan osoite esitetään kuvaa ympäröivän `<a>`-elementin kohdeosoitteena. Täysikokoisen kuvan näyttämisestä huolehtii WP LightBox 2 -lisäosa. Jos tuotteelle on määritetty useita kuvia, näytetään ensimmäinen kuva samaan tapaan kuin yhden kuvan tapauksessa. Lisäkuvat näytetään pienempinä versioina pääkuvan alapuolella ja klikkaamalla näitä kuvia pääkuva vaihtuu.

Sisältöosion tekstiosa koostuu vapaan tekstikentän sisällöstä sekä hinnasta. Tekstikenttä on määritetty Wysiwyg-editorina, jolloin WordPress lisää kirjoitettuun tekstiin tarvittavat HTML-merkkaukset. Jos tuotteelle on määritetty hinta, tehdään tarkastus, onko tuote alennuksessa. Tähän on määritetty funktio `onAlessa`, joka tarkastaa onko alennushintaa määritetty ja onko alennukselle määritetty aikaväli voimassa. Tämän tarkastuksen perusteella näytetään joko tuotteen hinta tai alennettu hinta sekä alkuperäinen hinta yliviivattuna.

Lisätieto-osion lähdekoodi on kuvattu liitteessä 1. Osio koostuu maksimissaan kolmesta alaosiosta, jotka ovat mitat, valmistaja ja kysy tuotteesta. Mitat ja valmistaja näytetään jos instanssille on määritetty kyseiset tiedot. Valmistaja noudeetaan WP_Queryllä, joka tekee sisältökyselyn instansseille, jotka on liitetty tietyllä yhteystunnuksella käsiteltävään instanssiin. Tämän silmukan sisällä ollaan valmistaja-sisältötyypin instanssin kontekstissa, jolloin tietokenttiä voidaan kutsua samalla tavalla kuin itse valmistaja-instanssin sivulla. Viimeisessä osiossa esitetään options-rajapinnan avulla tallennetut yhteystiedot sekä yhteydenottolomake.

Sivun viimeisessä osassa esitetään lisätuotteita. Lisätuotteet haetaan WP_Querylla, joka etsii tuotteet-sisältötyypin instansseja, joilla on yksi tai useampi käsiteltävän instanssin tuoteryhmistä. Tulostuloksen järjestys satunnaistetaan, jotta kysely ei näyttäisi tuoteryhmiltään identtisillä tuotteilla samoja lisätuotteita. Lisätuotteet arvotaan jokaisella sivulatauksella uudestaan. Tulostulostulo on rajattu maksimissaan kolmeen tuotteeseen. Lisätuotteista esitetään kuva, nimi ja erityishuomautus, jos tuote on alennuksessa. Alennuksen tarkastus tehdään samalla onAlessa-funktiolla kuten sisältöosassa.

5.3 Tuotelistaus

Kaikkien tuoteryhmien listaus toteutetaan sivupohjaan tuoteryhmat.php, joka on valinnainen sivupohja sivu-sisältötyypille. Sivupohja otetaan käyttöön valitsemalla sivupohja sivun asetukset -osassa muokkausnäkyssä. Tuotelistaus koostuu tuoteryhmäosioista, jotka esitetään allekkain. Tuoteryhmäosa sisältää tuoteryhmän nimen, linkin tuoteryhmäkohtaiselle sivulle sekä kolme uusinta tuotetta. Jos tuoteryhmällä ei ole tuotteita, sitä ei näytetä listauksessa.

Tuoteryhmien termit, jotka eivät ole tyhjiä, haetaan get_terms-funktiolla, joka palauttaa termit taulukkoon oliomuodossa. Termit käsitellään foreach-silmukalla, jonka sisällä luodaan kullakin iteraatiolla WP_Query-kysely, joka etsii kolme uusinta tuotetta, joihin on liitetty käsiteltävä tuoteryhmätermi. Termistä esitetään

nimi sekä arkistosivun (archive) linkki, joka haetaan termin id:n perusteella. Arkistosivulla listataan kaikki tuoteryhmään liitetyt tuotteet. Niiden näyttämiseen tarvitaan oma sivupohja.

Tuoteryhmien sisäinen tuotelistaus toteutetaan sivupohjaan taxonomy-tuoteryhma.php, jolloin sitä käytetään kaikissa tuoteryhmä-taksonomian termien näkymissä. Sivupohjan nimeäminen WordPressin dokumentaatiossa määriteltyjen nimeämiskäytäntöjen mukaan ottaa sen automaattisesti käyttöön. Tuotelistaus sisältää tuoteryhmän nimen, alatermit, järjestyksen vaihtamisen sekä tuoteryhmään liitetyt tuotteet.

Tuotteiden esittäminen sarakkeina, järjestyksen manipulointi ja alatuoteryhmien suodatus on toteutettu Isotope-javascript-kirjastolla. Tuoteryhmän sivulla on listattu yhteen näkymään kaikki tuoteryhmään liitetyt tuotteet. Klikkaamalla alatuoteryhmän nimeä listauksesta poistetaan ne tuotteet, jotka eivät sisällä valittua termiä. Suodatus tapahtuu ilman sivun uudelleenlatausta. Tuotteiden järjestys voi olla lisäysperusteinen, aakkosellinen, hintaperusteinen tai alennetut korostava. Järjestyksen vaihto tapahtuu myös ilman sivun uudelleenlatausta.

Tuotteet haetaan WP_Query-kyselyllä, joka palauttaa kaikki tuote-sisältötyypin instanssit, joihin on liitetty käsiteltävä tuoteryhmä. Tuotteiden uloimmaiseen <div>-elementtiin liitetään attribuutteja, joita hyödynnetään Isotope-skriptillä. Elementin luokkiin (class) lisätään kaikki tuotteeseen liitetyt tuoteryhmät. Mukautettuun attribuuttiin data-aakkosellinen tallennetaan tuotteen nimi, josta on poistettu välilyönnit. Attribuuttiin data-ale tallennetaan tieto siitä onko tuote alennuksessa. Mahdollisia arvoja ovat 1 ja 0.

Alatuoteryhmien nimet haetaan get_term_children-funktiolla, jonka avulla saadaan taulukko, joka sisältää tuoteryhmän alatermit. Jos käsiteltävälle tuoteryhmälle on määritetty alatuoteryhmiä, taulukko käsitellään foreach-silmukassa, jossa termit tulostetaan ympäröitynä <a>-elementtiin. Elementin data-filter-attribuutin arvona on termin nimi. Järjestyksen valinnan käyttöliittymä toteutetaan Bootstrap-tyylikehykseen perustuvalla pudotusvalikolla. Järjestysvalintoihin liitetään attribuutti data-sort-by ja kullekin valinnalle yksilöllinen tunnus.

Listauksen hallinnan interaktiivisuus toteutetaan Isotope-luokan instanssilla, joka hyödyntää aiemmin määritettyjä attribuutteja. Skriptin avulla tuotteet jaetaan kolmeksi sarakkeeksi ja järjestysperusteiden kriteerit määritetään. Alustuksessa järjestystä ei muuteta vaan järjestys on WP_Query:n määrittämä. Klikkaamalla alatuoteryhmää muutetaan Isotope-instanssin filter-attribuuttia ja tuotelistaa suodatetaan. Järjestyksen muuttaminen tapahtuu manipuloimalla Isotope-instanssin sortBy-attribuuttia, jonka muuttaminen järjestää tuotelistauksen määritettyjen kriteerien mukaisesti.

5.4 Valmistajasivu

Valmistajasivu tehdään single-tuotemerkki.php-sivupohjaan, jolloin se on käytössä valmistaja-sisältötyypin instansseissa. Valmistajasivun sisältöosa muodostuu neljästä osiosta, jotka ovat valmistajan kuvaus, tuotteet, outlet-tuotteet ja kuvat. Näistä vain ensimmäinen on käytössä aina. Relaatio-osiot näytetään jos valmistajalle on liitetty osioihin kuuluvia sisältötyyppien instansseja.

Valmistajan kuvaus muodostuu logosta, tekstiosasta ja linkistä valmistajan sivustolle. Nämä tiedot saadaan suoraan kontekstiriippuvasta silmukasta, sillä ne koskevat käsiteltävää instanssia. Linkkiin lisätään attribuutti target, jolle annetaan arvoksi _blank, jolloin linkki aukeaa uuteen välilehteen.

Relaatio-osio sisältää kolme WP_Query-kyselyä. Kyselyt käyttävät jokaista Posts 2 Posts -lisäosalle määritettyä relaatiotyyppiä. Tuotteet esitetään ilman jaottelua tuoteryhmiin. Muuten listaus mukailee tuoteryhmän sisäistä listausta. Valmistaja-kohtaista tuotelistan järjestystä on mahdollista muuttaa kuten tuoteryhmäsivulla. WP_Query laskee löydetyn tulosjoukon lukumäärän ja kokonaismäärä ilmoitetaan osion otsikon vieressä.

Outlet-tuotteiden listausnäköymä mukailee tuotelistausta. Järjestysperusteina on lisäksi hinta, joka yleisesti liitetään outlet-tuotteisiin. Hinta on mahdollista liittää myös tavallisiin tuotteisiin, mutta oletuksena sen liittäminen tilattaviin tuotteisiin

on harvinaisempaa, koska tilattavien tuotteiden hinta vaihtelee esimerkiksi materiaalivalintojen vuoksi. Tästä syystä tuotteille ei tarjota vastaavaa järjestysmahdollisuutta.

Valmistajasivun viimeisenä osiona on kuvastot. Kuvastoille ei näytetä järjestysmahdollisuuksia, koska kuvastoja on oletuksena vähän. Kuvastosta on suora linkki PDF-muotoiseen tiedostoon. Kuvastot järjestetään listaukseen julkaisujärjestyksessä.

5.5 Hakutoiminto

Hakutoiminto toteutetaan Search Everything -lisäosalla, joka laajentaa ytimen hakutoiminnallisuutta ottamalla huomioon instanssien metatiedot eli mukautetuilla kentillä syötetyt tiedot. Lisäosan avulla hakulauseen jäsenet voidaan korostaa hakutuloksissa. Lisäosa ei vaikuta hakukentän ulkoasuun.

Hakukenttään kirjoitetaan hakulauseke, joka lähetetään WordPressin käsiteltäväksi. WordPress ohjaa käyttäjän hakutulossivulle ja kutsuu teeman search.php-tiedostoa. Jos search.php-tiedostoa ei ole määritetty käytetään index.php-tiedostoa. Hakutulokset käsitellään kontekstiriippuvaisella silmukalla ja tilanne, jossa hakutuloksia ei löydetty, voidaan käsitellä samassa sivupohjassa. (Hedengren 2014, 92-93.)

Toteutuksen ongelmana oli näyttää hakutuloksista relevantit tiedot, sillä hakutulos saattoi olla uutinen, valmistaja, tuote, outlet-tuote tai kuvasto. Kontekstiriippuvaisen silmukan iteraatioissa käsiteltävän tuloksen sisältötyyppi selvitetään ytimen get_post_type-funktiolla, joka palauttaa sisältötyypin tunnuksen. Kun sisältötyyppi tiedetään, voidaan kutsua sopivia sisältötyyppikohtaisia kenttiä.

Hakutulokset

	Apollo- lipastot	Apollo- lipastot ovat korkealaatuisia lipastoja , joissa runko ja etulevyt...	Tuotteet
	Jugend- lipasto	Niemen Tehtaiden poistuvan Jugend-sarjan kaunis Jugend- lipasto .	Outlet
	Fantasia- lipasto	Ylivieskan Puuseppien valmistama Fantasia- lipasto – nyt edullisesti	Outlet

Kuva 5. Hakutuloksien esittäminen.

Hakutulosten muotoilu esitetään kuvassa 5. Tuotteista ja outlet-tuotteista näytetään hakutuloksissa kuva, otsikko ja katkelma kuvauksesta sekä kaikissa hakutuloksissa näkyvä sisältötyyppi. Kuvastoista näytetään kansikuva, otsikko ja ala-otsikko. Valmistajista näytetään logo, otsikko ja katkelma kuvauksesta.

6 POHDINTA

Opinnäytetyön tavoitteena oli suunnitella, toteuttaa ja hyödyntää sisältörakenteita WordPress-ympäristössä. Asetetut tavoitteet täytettiin ja toimeksiantaja oli tyytyväinen lopputulokseen. Sisällönrakenteiden hyödyntämisvaiheen jälkeen syötettiin sisältö, minkä jälkeen sivusto julkaistiin joulukuussa 2014.

Raportointi paljasti haasteeksi aiheen laajuuden, sillä sisältörakenteita tarkastellaan kolmesta näkökulmasta. En pitänyt tarkempaa rajausta mielekkäänä, jotta sisältörakenteita voidaan käsitellä ehjänä kokonaisuutena. Käsiteltävät näkökulmat liittyvät kiinteästi toisiinsa ja esimerkiksi toteutuksessa tehtyjen valintojen ymmärtäminen vaatii suunnitteluratkaisujen läpikäyntiä. Valittu rajausta merkitsi osittain hyvin pintapuolista asioiden tarkastelua.

Työn ongelmakohtana oli suunnitteluvaiheessa arvioida sisältörakennerratkaisujen tarkoituksenmukaisuus pitkällä aikavälillä. Suunniteltu sisältörakenne osoittautui toimivaksi toteutus- ja hyödyntämisvaiheissa. Ainut ongelma ylläpidettävyyden puolesta on tuotteiden ja outlet-tuotteiden erottaminen omiksi sisältötyypeikseen. Nämä olisi voitu toteuttaa samassa sisältötyypissä ja ero olisi voitu talmentaa esimerkiksi tuotetyyppi-taksonomian avulla. Näin sisällönhakulogiikkaa olisi voitu yksinkertaistaa.

Työskentely WordPressin parissa sujui hyvin, sillä minulla on runsaasti kokemusta WordPress-projekteista. Aktiivinen yhteisö ja verkon keskustelupalstat auttoivat spesifeissä ongelmatapauksissa. Käytetty kirjallisuus määritteli hyvät käytännöt yleisellä tasolla.

Jatkokehityskohteena on sisältötyyppien sisäisten linkitysten hallinta ja hyödyntäminen. Sisältötyyppien sisäiset linkitykset mahdollistaisivat esimerkiksi tuotteiden liittämisen toisiin tuotteisiin, jolloin tuotesivulla voidaan esittää manuaalisesti valittuja tuotteeseen liittyviä tuotteita. Tämä mahdollistaisi tuoteryhmät ylittävät yhteydet esimerkiksi tietyn sohvan ja sohvapöydän välille.

Toinen jatkokehityskohde on kokoelmien hyödyntäminen. Kokoelmat sisältävät joukon instansseja sekä mahdollisesti muita kenttiä. Instanssit ovat tallennettuna viittauksina. WordPressin taksonomiat tarjoavat sisäänrakennetun tavan luoda kokoelmia, mutta hallintakäyttöliittymän kannalta ne eivät välttämättä ole helppo-käyttöisin ratkaisu. Kokoelmat mahdollistavat esimerkiksi yhteenkuuluvien tuotteiden esittämisen ryhmänä tai kampanjatuotteiden kokoamisen.

Työssä esitettyjä työkaluja ja suunnittelumenetelmiä voidaan hyödyntää pohjana verkkosivuprojekteissa, joihin kuuluu sisältörakenteiden suunnittelu ja toteutus. Opinnäytetyö esittää vaiheistetun prosessin sisältörakenteiden rakennukseen, joka lähtee tavoitteiden määrittelystä sekä kohdeyleisön analysoinnista ja päättyy suunniteltujen rakenteiden toteutukseen ja hyödyntämiseen. Opinnäytetyö esittää perusteet WordPressin sisällönhallintaominaisuuksien hyödyntämisestä.

LÄHTEET

- BuiltWith 2015. CMS Usage Statistics. Viitattu 31.3.2015 <http://trends.builtwith.com/cms>.
- Condon, E. 2015. What is ACF? Viitattu 28.3.2015 <http://www.advancedcustomfields.com/resources/what-is-acf/>.
- Coyier, C. & Starr, J. 2014. Digging into WordPress. Chris Coyier and Jeff Starr.
- Crawford, S. 2015a. Components Used in Information Architecture. Viitattu 4.4.2015 <http://computer.howstuffworks.com/information-architecture4.htm>.
- Crawford, S. 2015b. Designing an Information Architecture. Viitattu 4.4.2015 <http://computer.howstuffworks.com/information-architecture5.htm>.
- Gillham, B. 2010. Case Study Research Methods. London: Continuum International Publishing.
- Hedengren, T. 2014. Smashing WordPress: Beyond the Blog. 4th Edition. West Sussex: John Wiley & Sons.
- Król, K. & Silver, A. 2013. WordPress 3.7 Complete. 3rd Edition. Birmingham: Packt Publishing.
- Laja, P. 2013. Getting The Website Information Architecture Right: How to Structure Your Site for Optimal User Experiences. Viitattu 26.3.2015 <http://conversionxl.com/website-information-architecture-optimal-user-experience/>.
- Lim, W. 2012. A Beginner's Guide to Wireframing. Viitattu 4.4.2015 <http://webdesign.tutsp-lus.com/articles/a-beginners-guide-to-wireframing--webdesign-7399>.
- Lukka, K. 2014. Konstruktiivinen tutkimusote. Viitattu 31.3.2015. <https://metodix.wordpress.com/2014/05/19/lukka-konstruktiivinen-tutkimusote/>.
- McCollin, R. 2012. WordPress Pushing the Limits. West Sussex: John Wiley & Sons.
- Ratmayake. R. 2013. WordPress Web Application Development. Birmingham: Packt Publishing.
- Rueping, A. 2010. Wiley Software Patterns: Where Code and Content Meet: Design Patterns for Web Content Management and Delivery, Personalisation and User Participation. West Sussex: John Wiley & Sons.
- TecEd 2015. Information Architecture as Part of UCD. Viitattu 5.4.2015 <http://teced.com/services/user-interface-design/information-architecture-as-part-of-ucd/>.
- Usability.gov 2015a. Information Architecture Basics. Viitattu 4.4.2015 <http://www.usability.gov/what-and-why/information-architecture.html>.
- Usability.gov 2015b. Organization Structures. Viitattu 4.4.2015 <http://www.usability.gov/how-to-and-tools/methods/organization-structures.html>.
- Wordpress.org 2015a. Database Description. Viitattu 28.3.2015 https://codex.wordpress.org/Database_Description.
- Wordpress.org 2015b. Taxonomies. Viitattu 28.3.2015 <https://codex.wordpress.org/Taxonomies>.
- Wordpress.org 2015c. Function Reference/register post type. Viitattu 28.3.2015 https://codex.wordpress.org/Function_Reference/register_post_type.

Wordpress.org 2015d. Roles and Capabilities. Viitattu 28.3.2015 https://codex.wordpress.org/Roles_and_Capabilities.

Wordpress.org 2015e. Using Image and File Attachments. Viitattu 28.3.2015 https://codex.wordpress.org/Using_Image_and_File_Attachments.

Wordpress.org 2015f. Image Size and Quality. Viitattu 28.3.2015 https://codex.wordpress.org/Image_Size_and_Quality.

Wordpress.org 2015g. Class Reference/WP Query. Viitattu 28.3.2015 https://codex.wordpress.org/Class_Reference/WP_Query.

Wordpress.org 2015h. Using Permalinks. Viitattu 28.3.2015 https://codex.wordpress.org/Using_Permalinks.

Wordpress.org 2015i. Template Hierarchy. Viitattu 5.4.2015 https://codex.wordpress.org/Template_Hierarchy.

Tuotesivujen lisätieto-osion toteutus

```

1 <div class="single-tuote-tabs">
2
3 <!-- Välilehtien navigaatio -->
4 <ul class="nav nav-tabs" id="meta-tabs">
5 <?php
6 // Lisätään Mitat välilehtivalikkoon, jos mitat on määritetty.
7 // Mitat-välilehti on oletuksena auki.
8 $mitat = get_field('tuote-mitat');
9 if(!empty($mitat)) {
10 echo '<li class="active"><a href="#mitat" data-toggle="tab">Mitat</a></li>';
11 }
12
13 // Lisätään Valmistaja välilehtivalikkoon, jos valmistaja on määritetty.
14 // Välilehti on auki, jos mittoja ei ole.
15 $connected = new WP_Query(array(
16 'connected_type' => $connection_type,
17 'connected_items' => get_queried_object(),
18 'nopaging' => true,
19 ));
20 if ($connected->have_posts()) {
21 $valmistaja = true;
22 $valmistaja_aktiivisuus = '';
23 if(empty($mitat)){
24 $valmistaja_aktiivisuus = 'class="active"';
25 }
26 echo '<li ' . $valmistaja_aktiivisuus . '><a href="#valmistaja" ' .
27 'data-toggle="tab">Valmistaja</a></li>';
28 }
29
30 // Lisätään Kysy tuotteesta välilehtivalikkoon.
31 // Välilehti on auki jos muita välilehtiä ei ole.
32 $kysy_aktiivisuus = '';
33 if(empty($mitat) && !$valmistaja){
34 $kysy_aktiivisuus = 'class="active"';
35 }
36 echo '<li ' . $kysy_aktiivisuus . '><a href="#kysy" ' .
37 'data-toggle="tab">Kysy tuotteesta</a></li>';
38 ?>
39 </ul>
40
41 <div class="tab-content">
42
43 <!-- Mitat-välilehti -->
44 <?php if(!empty($mitat)) : ?>
45 <div class="tab-pane active" id="mitat">
46 <div class="tab-padding">
47 <p><?php echo $mitat; ?></p>
48 </div>
49 </div>
50 <?php endif; ?>
51
52 <!-- Valmistaja-välilehti -->
53 <?php if($valmistaja) : ?>
54 <div class="tab-pane <?php if(empty($mitat)){echo "active";} ?>" id="valmistaja">
55 <div class="tab-padding">
56 <?php
57 $connected = new WP_Query(array(
58 'connected_type' => $connection_type,
59 'connected_items' => get_queried_object(),
60 'nopaging' => true,
61 ));
62 if($connected->have_posts()) :

```

```

63         while ( $connected->have_posts() ) : $connected->the_post(); ?>
64         <div class="row">
65             <div class="col-sm-3 center">
66                 <?php
67                 $title = get_the_title();
68                 $logo = get_field('logo');
69                 $url = get_permalink();
70                 if(!empty($logo)) {
71                     $image = wp_get_attachment_image_src($logo, "attachment");
72                     echo '<a href="' . $url . '"></a>';
74                 } else {
75                     echo '<a href="' . $url . '"> . $title . '</a>';
76                 } ?>
77             </div>
78             <div class="col-sm-9">
79                 <?php echo custom_field_excerpt(); ?>
80                 <a href="<?php echo $url; ?>">Lue lisää</a>
81             </div>
82         </div>
83         <?php endwhile; ?>
84         <?php wp_reset_postdata(); ?>
85     <?php endif; ?>
86 </div>
87 </div>
88 <?php endif; ?>
89
90 <!-- Kysy lisää -välilehti -->
91 <div class="tab-pane <?php echo $kysy_aktiivisuus; ?>" id="kysy">
92     <div class="tab-padding">
93         <div class="row">
94             <div class="col-sm-6">
95                 <?php echo do_shortcode(['contact-form-7 id="74" title="Kysy tuotteesta"]'); ?>
96             </div>
97             <div class="col-sm-6">
98                 <p><?php the_field('kysy', 'option'); ?></p>
99                 <p>
100                     <span class="glyphicon glyphicon-phone-alt ikoni"></span>
101                     <?php the_field('puhelin', 'option'); ?>
102                 </p>
103                 <p>
104                     <span class="glyphicon glyphicon-phone ikoni"></span>
105                     <?php the_field('matkapuhelin', 'option'); ?>
106                 </p>
107                 <p>
108                     <span class="glyphicon glyphicon-envelope ikoni"></span>
109                     <?php the_field('sahkoposti', 'option'); ?>
110                 </p>
111             </div>
112         </div>
113     </div>
114 </div>
115 </div>
116 </div>
117
118 <script>
119     $(function () {
120         $('#meta-tabs a:first').tab('show');
121     });
122 </script>

```